Utf manual page - Tcl Library Procedures

tcl.tk/man/tcl/TclLib/Utf.htm

NAME

Tcl_UniChar, Tcl_UniCharToUtf, Tcl_UtfToUniChar, Tcl_UniCharToUtfDString, Tcl_UtfToUniCharDString, Tcl_UniCharLen, Tcl_UniCharNcmp, Tcl_UniCharNcasecmp, Tcl_UniCharCaseMatch, Tcl_UtfNcmp, Tcl_UtfNcasecmp, Tcl_UtfCharComplete, Tcl_NumUtfChars, Tcl_UtfFindFirst, Tcl_UtfFindLast, Tcl_UtfNext, Tcl_UtfPrev, Tcl_UniCharAtIndex, Tcl_UtfAtIndex, Tcl_UtfBackslash — routines for manipulating UTF-8 strings

SYNOPSIS

#include <tcl.h>
typedef ... **Tcl_UniChar**;
int
Tcl_UniCharToUtf(*ch, buf*)
int
Tcl_UtfToUniChar(*src, chPtr*)
char *
Tcl_UniCharToUtfDString(*uniStr, uniLength, dsPtr*)
Tcl_UniChar *
Tcl_UtfToUniCharDString(*src, length, dsPtr*)
int
Tcl_UniCharLen(*uniStr*)
int
Tcl_UniCharNcmp(*ucs, uct, numChars*)
int
Tcl_UniCharNcasecmp(*ucs, uct, numChars*)
int
Tcl_UniCharCaseMatch(*uniStr, uniPattern, nocase*)
int
Tcl_UtfNcmp(*cs, ct, numChars*)
int
Tcl_UtfNcasecmp(*cs, ct, numChars*)
int
Tcl_UtfCharComplete(*src, length*)
int
Tcl_NumUtfChars(*src, length*)
const char *
Tcl_UtfFindFirst(*src, ch*)
const char *
Tcl_UtfFindLast(*src, ch*)

const char *
Tcl_UtfNext(*src*)
const char *
Tcl_UtfPrev(*src, start*)
Tcl_UniChar
Tcl_UniCharAtIndex(*src, index*)
const char *
Tcl_UtfAtIndex(*src, index*)
int
Tcl_UtfBackslash(*src, readPtr, dst*)

ARGUMENTS

char *buf (out)
Buffer in which the UTF-8 representation of the Tcl_UniChar is stored. At most **TCL_UTF_MAX** bytes are stored in the buffer.

int ch (in)
The Unicode character to be converted or examined.

Tcl_UniChar *chPtr (out)
Filled with the Tcl_UniChar represented by the head of the UTF-8 string.

const char *src (in)
Pointer to a UTF-8 string.

const char *cs (in)
Pointer to a UTF-8 string.

const char *ct (in)
Pointer to a UTF-8 string.

const Tcl_UniChar *uniStr (in)
A null-terminated Unicode string.

const Tcl_UniChar *ucs (in)
A null-terminated Unicode string.

const Tcl_UniChar *uct (in)
A null-terminated Unicode string.

const Tcl_UniChar *uniPattern (in)
A null-terminated Unicode string.

int length (in)
The length of the UTF-8 string in bytes (not UTF-8 characters). If negative, all bytes up to the first null byte are used.

int uniLength (in)
The length of the Unicode string in characters. Must be greater than or equal to 0.

Tcl_DString *dsPtr (in/out)
A pointer to a previously initialized **Tcl_DString**.

unsigned long numChars (in)
The number of characters to compare.

const char *start (in)
Pointer to the beginning of a UTF-8 string.

int index (in)
The index of a character (not byte) in the UTF-8 string.

int *readPtr (out)
If non-NULL, filled with the number of bytes in the backslash sequence, including the backslash character.

char *dst (out)
Buffer in which the bytes represented by the backslash sequence are stored. At most **TCL_UTF_MAX** bytes are stored in the buffer.

int nocase (in)
Specifies whether the match should be done case-sensitive (0) or case-insensitive (1).

DESCRIPTION

These routines convert between UTF-8 strings and Tcl_UniChars. A Tcl_UniChar is a Unicode character represented as an unsigned, fixed-size quantity. A UTF-8 character is a Unicode character represented as a varying-length sequence of up to **TCL_UTF_MAX** bytes. A multibyte UTF-8 sequence consists of a lead byte followed by some number of trail bytes.
TCL_UTF_MAX is the maximum number of bytes that it takes to represent one Unicode character in the UTF-8 representation.

Tcl_UniCharToUtf stores the Tcl_UniChar *ch* as a UTF-8 string in starting at *buf*. The return value is the number of bytes stored in *buf*.

Tcl_UtfToUniChar reads one UTF-8 character starting at *src* and stores it as a Tcl_UniChar in **chPtr*. The return value is the number of bytes read from *src*. The caller must ensure that the source buffer is long enough such that this routine does not run off the end and dereference non-existent or random memory; if the source buffer is known to be null-terminated, this will not happen. If the input is not in proper UTF-8 format, **Tcl_UtfToUniChar** will store the first byte of *src* in **chPtr* as a Tcl_UniChar between 0x80 and 0xFF and return 1.

Tcl_UniCharToUtfDString converts the given Unicode string to UTF-8, storing the result in a previously initialized **Tcl_DString**. You must specify *uniLength*, the length of the given Unicode string. The return value is a pointer to the UTF-8 representation of the Unicode string. Storage for the return value is appended to the end of the **Tcl_DString**.

Tcl_UtfToUniCharDString converts the given UTF-8 string to Unicode, storing the result in the previously initialized **Tcl_DString**. In the argument *length*, you may either specify the length of the given UTF-8 string in bytes or “-1”, in which case

Tcl_UtfToUniCharDString uses **strlen** to calculate the length. The return value is a pointer to the Unicode representation of the UTF-8 string. Storage for the return value is appended to the end of the **Tcl_DString**. The Unicode string is terminated with a Unicode null character.

Tcl_UniCharLen corresponds to **strlen** for Unicode characters. It accepts a null-terminated Unicode string and returns the number of Unicode characters (not bytes) in that string.

Tcl_UniCharNcmp and **Tcl_UniCharNcasecmp** correspond to **strncmp** and **strncasecmp**, respectively, for Unicode characters. They accept two null-terminated Unicode strings and the number of characters to compare. Both strings are assumed to be at least *numChars* characters long. **Tcl_UniCharNcmp** compares the two strings character-by-character according to the Unicode character ordering. It returns an integer greater than, equal to, or less than 0 if the first string is greater than, equal to, or less than the second string respectively. **Tcl_UniCharNcasecmp** is the Unicode case insensitive version.

Tcl_UniCharCaseMatch is the Unicode equivalent to **Tcl_StringCaseMatch**. It accepts a null-terminated Unicode string, a Unicode pattern, and a boolean value specifying whether the match should be case sensitive and returns whether the string matches the pattern.

Tcl_UtfNcmp corresponds to **strncmp** for UTF-8 strings. It accepts two null-terminated UTF-8 strings and the number of characters to compare. (Both strings are assumed to be at least *numChars* characters long.) **Tcl_UtfNcmp** compares the two strings character-by-character according to the Unicode character ordering. It returns an integer greater than, equal to, or less than 0 if the first string is greater than, equal to, or less than the second string respectively.

Tcl_UtfNcasecmp corresponds to **strncasecmp** for UTF-8 strings. It is similar to **Tcl_UtfNcmp** except comparisons ignore differences in case when comparing upper, lower or title case characters.

Tcl_UtfCharComplete returns 1 if the source UTF-8 string *src* of *length* bytes is long enough to be decoded by **Tcl_UtfToUniChar**/**Tcl_UtfNext**, or 0 otherwise. This function does not guarantee that the UTF-8 string is properly formed. This routine is used by procedures that are operating on a byte at a time and need to know if a full Tcl_UniChar has been seen.

Tcl_NumUtfChars corresponds to **strlen** for UTF-8 strings. It returns the number of Tcl_UniChars that are represented by the UTF-8 string *src*. The length of the source string is *length* bytes. If the length is negative, all bytes up to the first null byte are used.

Tcl_UtfFindFirst corresponds to **strchr** for UTF-8 strings. It returns a pointer to the first occurrence of the Tcl_UniChar *ch* in the null-terminated UTF-8 string *src*. The null terminator is considered part of the UTF-8 string.

Tcl_UtfFindLast corresponds to **strrchr** for UTF-8 strings. It returns a pointer to the last occurrence of the Tcl_UniChar *ch* in the null-terminated UTF-8 string *src*. The null terminator is considered part of the UTF-8 string.

Given *src*, a pointer to some location in a UTF-8 string, **Tcl_UtfNext** returns a pointer to the next UTF-8 character in the string. The caller must not ask for the next character after the last character in the string if the string is not terminated by a null character. **Tcl_UtfCharComplete** can be used in that case to make sure enough bytes are available before calling **Tcl_UtfNext**.

Tcl_UtfPrev is used to step backward through but not beyond the UTF-8 string that begins at *start*. If the UTF-8 string is made up entirely of complete and well-formed characters, and *src* points to the lead byte of one of those characters (or to the location one byte past the end of the string), then repeated calls of **Tcl_UtfPrev** will return pointers to the lead bytes of each character in the string, one character at a time, terminating when it returns *start*.

When the conditions of completeness and well-formedness may not be satisfied, a more precise description of the function of **Tcl_UtfPrev** is necessary. It always returns a pointer greater than or equal to *start*; that is, always a pointer to a location in the string. It always returns a pointer to a byte that begins a character when scanning for characters beginning from *start*. When *src* is greater than *start*, it always returns a pointer less than *src* and greater than or equal to (*src* - **TCL_UTF_MAX**). The character that begins at the returned pointer is the first one that either includes the byte *src[-1]*, or might include it if the right trail bytes are present at *src* and greater. **Tcl_UtfPrev** never reads the byte *src[0]* nor the byte *start[-1]* nor the byte *src[-***TCL_UTF_MAX***-1]*.

Tcl_UniCharAtIndex corresponds to a C string array dereference or the Pascal Ord() function. It returns the Tcl_UniChar represented at the specified character (not byte) *index* in the UTF-8 string *src*. The source string must contain at least *index* characters. Behavior is undefined if a negative *index* is given.

Tcl_UtfAtIndex returns a pointer to the specified character (not byte) *index* in the UTF-8 string *src*. The source string must contain at least *index* characters. This is equivalent to calling **Tcl_UtfToUniChar** *index* times. If a negative *index* is given, the return pointer points to the first character in the source string.

Tcl_UtfBackslash is a utility procedure used by several of the Tcl commands. It parses a backslash sequence and stores the properly formed UTF-8 character represented by the backslash sequence in the output buffer *dst*. At most **TCL_UTF_MAX** bytes are stored in the buffer. **Tcl_UtfBackslash** modifies **readPtr* to contain the number of bytes in the backslash sequence, including the backslash character. The return value is the number of bytes stored in the output buffer.

See the **Tcl** manual entry for information on the valid backslash sequences. All of the sequences described in the Tcl manual entry are supported by **Tcl_UtfBackslash**.